Introduction to Machine Learning

Nusrat Ansari

Module	Content		Hours
1	**Introduction to Machine Learning**		03
	1.1	Machine Learning, Types of Machine Learning- Supervised, unsupervised and reinforcement, Issues in Machine Learning, Application of Machine Learning, Steps in developing a Machine Learning Application.	
	1.2	Training Error, Generalization error, Overfitting, Underfitting, Bias Variance trade-off	
2	**Learning with Regression**		06
	2.1	Learning with Regression: Linear Regression, Multivariate Linear Regression, Logistic Regression.	

		Algorithm	
5	**Data Engineering**		**04**
	5.1	Introduction to Data Engineering,Data Ingestion: Techniques and Best Practices,Data Storage and Management: Data Lakes, Data Warehouses, Data Processing Pipelines.	
	5.2	Lamda Architecture, Batch Processing, Stream Processing, Data Quality and Governance	

6	**Current Trends and tools used in ML**		**06**
	6.1	Introduction to Reinforcement learning (RL), Elements of RL, Model based, Temporal based	
	6.2	Machine Learning projects handle different types of data and tools in industries of Health Care & Agriculture	

	2.2	Performance Measures : Model evaluation and selection, Training, Testing and Validation Tests, Confusion Matrix & Basic Evaluation Metrics, Precision-recall.	
3	**Dimensionality Reduction**		**04**
	3.1	Curse of Dimensionality, Dimensionality Reduction Techniques, Principal Component Analysis, Linear Discriminant Analysis, Singular Value Decomposition.	
4	**Learning with Classification**		**06**
	4.1	Introduction to classification, Learning with Trees: Decision Trees, Constructing Decision Trees using Gini Index (Regression), Classification and Regression Trees (CART)	
	4.2	Introduction to Support Vector Machine (SVM), Hyperplane, Optimal decision boundary, Margins and support vectors, linear SVM, Nonlinear SVM, Kernelized SVM	
5	**Data Engineering**		**04**
	5.1	Introduction to Data Engineering,Data Ingestion: Techniques and Best Practices,Data Storage and Management: Data Lakes, Data Warehouses, Data Processing Pipelines.	

What is machine learning?

- A branch of **artificial intelligence**, concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data.

- As intelligence requires knowledge, it is necessary for the computers to acquire knowledge

What is machine learning?

- Machine Learning takes advantage of the ability of computer systems to learn from **correlations** hidden in the data; this ability can be further utilized by programming or developing intelligent and efficient Machine Learning algorithms.
- They use computational methods to “learn” information directly from data without relying on a predetermined equation as a model.
- As the number of samples available for learning increases, machine learning algorithms adaptively improve their performance.

What is machine learning?

- Machine Learning

Study of algorithms that improve their performance at some task with experience

- Optimize a performance criterion using example data or past experience.
- Role of Statistics: Inference from a sample
- Role of Computer science: Efficient algorithms to
  - Solve the optimization problem
  - Representing and evaluating the model for inference

What is machine learning?

The subfield of computer science that "gives computers the ability to learn without being explicitly programmed".
(Arthur Samuel, 1959)

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."
(Tom Mitchell, 1997)

What is machine learning?

Arthur Samuel (1959)

- ***Machine learning:*** **"Field of study that gives computers the ability to learn without being explicitly programmed"**
  - Samuels wrote a checkers playing program
    - Had the program play 10000 games against itself
    - Work out which board positions were good and bad depending on wins/losses

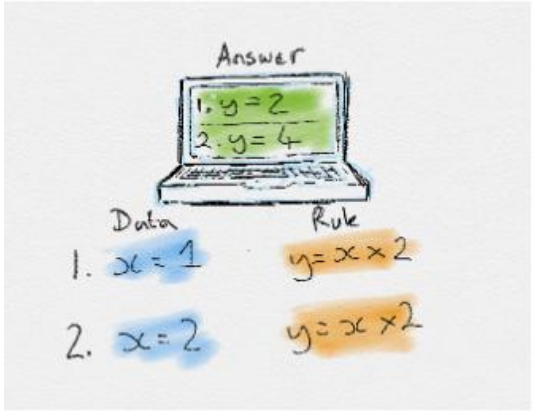
Tom Michel (1999)

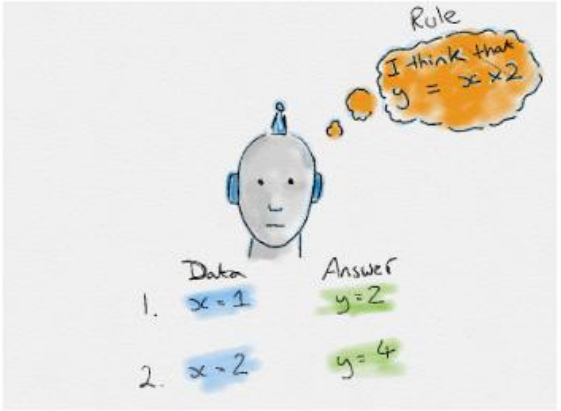
- ***Well posed learning problem:*** ***"*****A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."**
  - The checkers example,
    - E = 10000s games
    - T is playing checkers

History of AI and ML

- **Around 1960:**
  - –First wave of AI
  - –Inference: Given knowledge, “I” (AI) can make decisions like a “MAN”.
- **Around 1990:**
  - –Second wave of AI
  - –Learning: Given data, “I” can learn like a “MAN”.
- **Around 2020:**
  - –Third wave of AI
  - –Cyber-space: Given the internet, “I” can collect data and learn in a way different from “MAN”.

A simpler way





Traditional Programming vs Machine Learning

Traditional Programming

Machine Learning

Learning from Data

The world is driven by data.

- Germany's climate research centre generates 10 petabytes per year
- Google processes 24 petabytes per day
- The Large Hadron Collider produces 60 gigabytes per minute (~12 DVDs)
- There are over 50m credit card transactions a day in the US alone.
- **Data** is recorded from some real-world phenomenon.
- What might we want to do with that data?

- **Prediction**
- what can we **predict** about this phenomenon?
- **Description**
- how can we **describe/understand** this phenomenon in a new way?

- *How can we extract knowledge from data to help humans take decisions?*
- *How can we automate decisions from data?*
- *How can we adapt systems dynamically to enable better user experiences?*

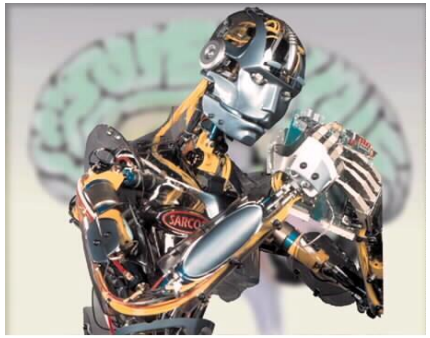
Write code to explicitly do the above tasks

Write code to make the computer ***learn*** how to do the tasks

Data – facts figures statistics data

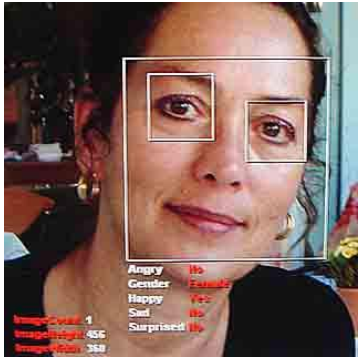
Data ---Information --- meaningful useful

What is Machine Learning?



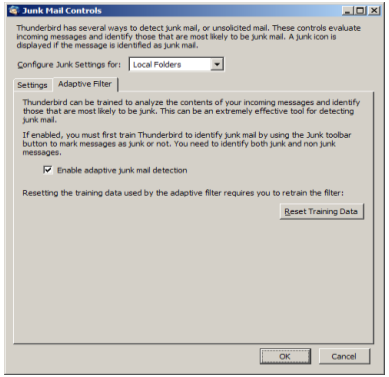
amazon.com

facebook

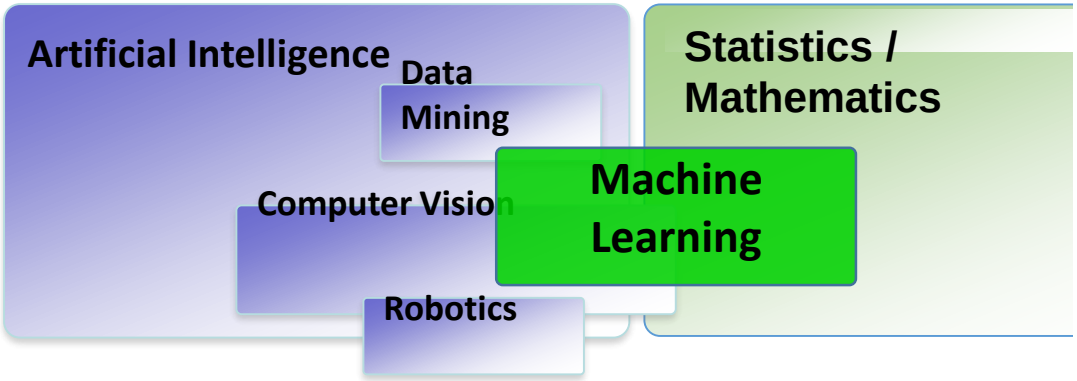


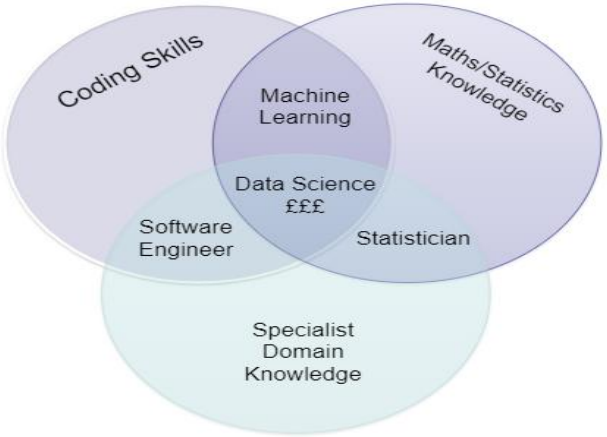
Microsoft

Lloyds TSB



ML: *Where does it fit? What is it not?*





ML in a Nutshell

- Tens of thousands of machine learning algorithms
- Hundreds new every year
- Every machine learning algorithm has three components:
  - **Representation**
  - **Evaluation**
  - **Optimization**

Representation

- Decision trees
- Sets of rules / Logic programs
- Instances
- Graphical models (Bayes/Markov nets)
- Neural networks
- Support vector machines
- Model ensembles
- Etc.

Evaluation

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence
- Etc.

Optimization

- Combinatorial optimization
  - E.g.: Greedy search
- Convex optimization
  - E.g.: Gradient descent
- Constrained optimization
  - E.g.: Linear programming

Definition of learning

Definition: A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasksT, as measured by P, improves with experience E.

Examples

Inducting Learning: Learning from experience

i) Handwriting recognition learning problem

- **Task T**: Recognising and classifying handwritten words within images
- **Performance P**: Percent of words correctly classified
- **Training experience E**: A dataset of handwritten words with given classifications

Machine - Learning - Tom Mitchell.

Definition of learning

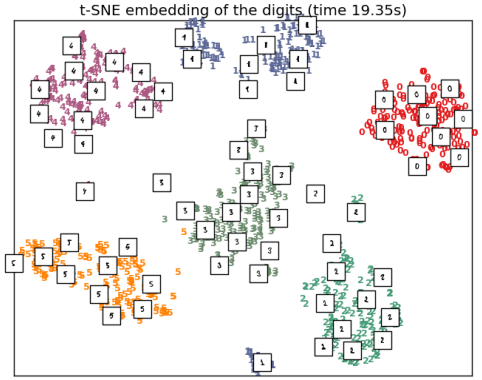
ii) A robot driving learning problem

- **Task T**: Driving on highways using vision sensors
- **Performance measure P:** Average distance traveled before an error
- **Training experience:** A sequence of images and steering commands recorded while observing a human driver

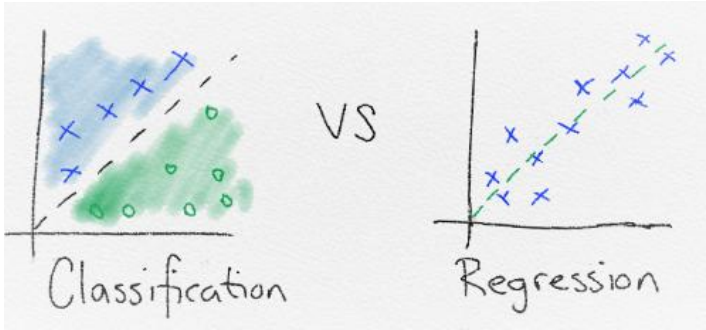
iii) A chess learning problem

- **Task T:** Playing chess
- **Performance measure P**: Percent of games won against opponents
- **Training experience E**: Playing practice games against itself

Types of Machine Learning



- **Supervised (inductive) learning**
  - Training data includes desired outputs-
  - Ex. Regression, Decision Tree, Random Forest, KNN, Logistic Regression, back propagation neural network
- **Unsupervised learning**
  - Training data does not include desired outputs-
  - Ex. clustering, dimensionality reduction and association rule learning..
- **Semi-supervised learning**
  - Training data includes a few desired outputs: **Speech Analysis,Internet Content Classification**
- **Reinforcement learning**
  - Rewards from sequence of actions



Input [temperature=**20**] -> *Model* -> Output = [visitors=**high**]

Input [temperature=**20**] -> *Model* -> Output = [visitors=**300**]

Types of Machine Learning

Supervised Learning : Teach the computer how to do something, then let it use it;s new found knowledge to do it

In supervised learning, the computer is provided with example inputs that are labeled with their desired outputs.

The purpose of this method is for the algorithm to be able to “learn” by comparing its actual output with the “taught” outputs to find errors, and modify the model accordingly.

Supervised learning therefore uses patterns to predict label values on additional unlabeled data.

Types of Machine Learning

Supervised

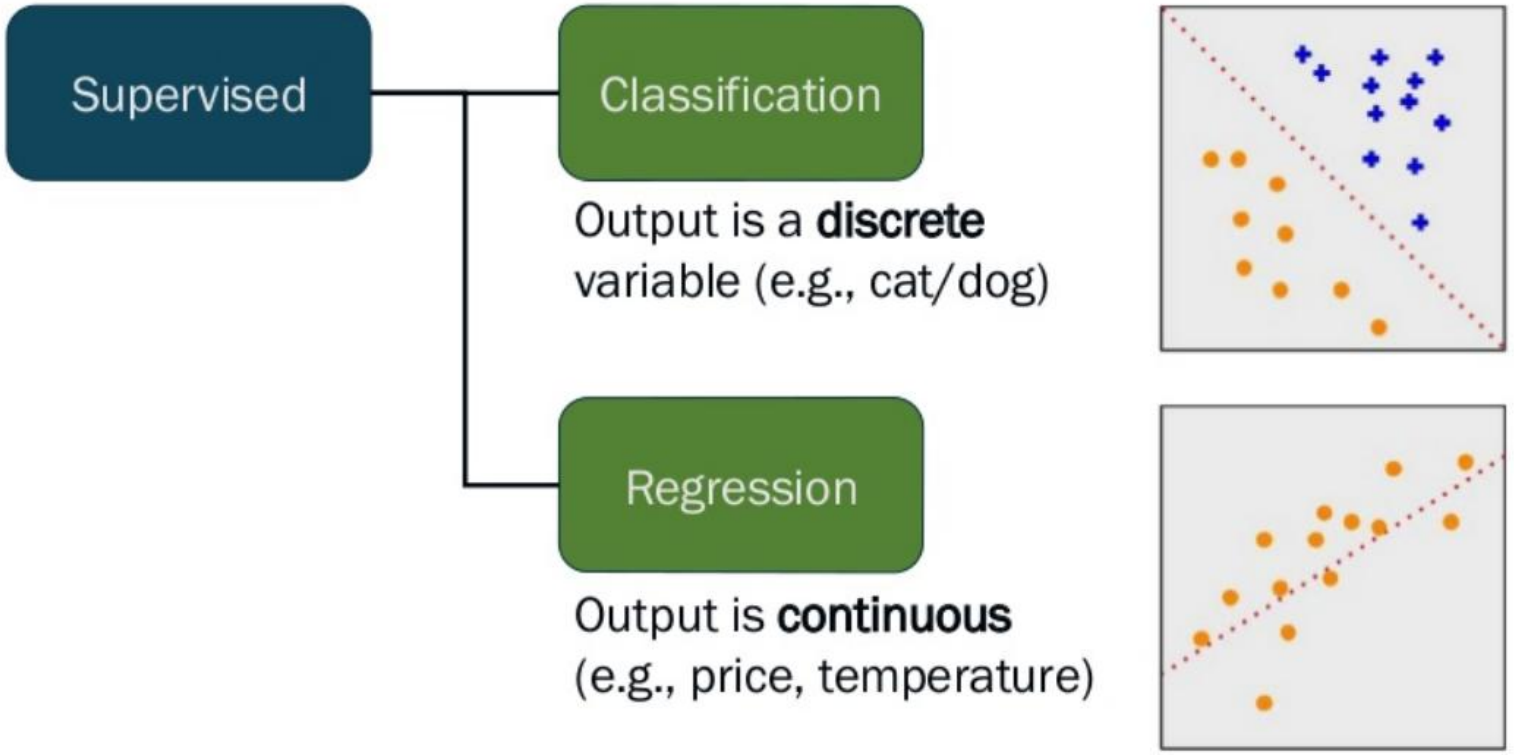
Learn through **examples** of which we know the desired output (what we want to predict).

Is this a cat or a dog?

Are these emails spam or not?

Predict the market value of houses, given the square meters, number of rooms, neighborhood, etc.

Types of Machine Learning



Types of Machine Learning

Unsupervised Learning: Let the computer learn how to do something, and use this to determine structure and patterns in data

In unsupervised learning, data is unlabeled, so the learning algorithm is left to find commonalities among its input data

Types of Machine Learning

Unsupervised

There is **no *desired* output**. Learn something about the data. *Latent* relationships.

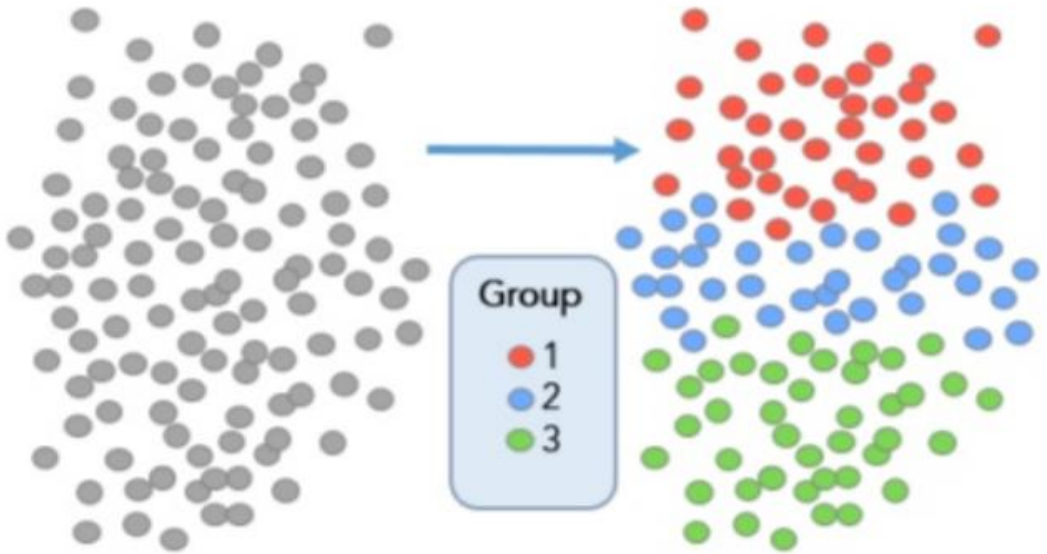
I have photos and want to put them in 20 groups.

I want to find anomalies in the credit card usage patterns of my customers.

Types of Machine Learning

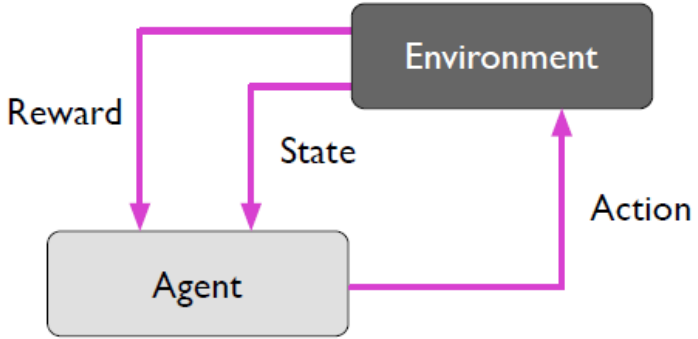
Unsupervised

Useful for learning structure in the data **(clustering)**, hidden correlations, reduce dimensionality, etc.



Types of Machine Learning

Reinforcement



An agent **interacts** with an **environment** and watches the result of the interaction.

Environment gives feedback via a positive or negative **reward signal**.

Reinforcement learning

- Reinforcement learning (RL) is important for “strategy learning”. It is useful for robotics, for playing games, etc.
- The well-known alpha-GO actually combined RL with deep learning, and was the first program that defeated human expert Go-players.
- In RL, a learner is called an agent. The point is to take a correct “action” for each environment “situation”.
- If there is a teacher who can tell the correct actions for all situations, we can use supervised learning.
- In RL, we suppose that the teacher only “rewards” or “punishes” the agent under some (not all) situations.

Reinforcement learning

- RL can find a “map” (a Q-table) that defines the relation between the situation set and the action set, so that the agent can get the largest reward by following this map.
- To play a game successfully, the computer can generate many different situations, and find a map between situation set and action set in such a way to win the game (with a high probability).
- Thus, even if there is no human opponent, a machine can improve its skill by playing with itself, using RL.
- Of course, if the machine has the honor to play many games with human experts, it can find the best strategy more “efficiently” without generating many “impossible” situations; or find good computer game players more acceptable to human.
- Examples: playing backgammon or chess, scheduling jobs, and controlling robot limbs

Semi Supervised Learning

In Semi supervised learning, some input data is labeled and some is unlabeled.

Classification

In machine learning, classification is the problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known.

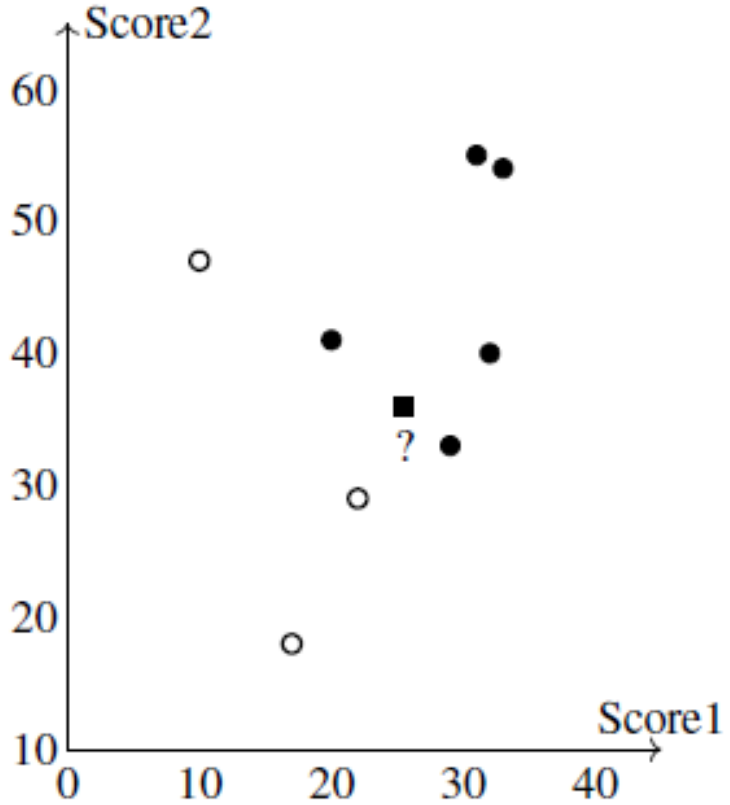
Example 1

Consider the following data:

Score1	29	22	10	31	17	33	32	20
Score2	43	29	47	55	18	54	40	41
Result	Pass	Fail	Fail	Pass	Fail	Pass	Pass	Pass

-training set of data.
-two attributes “Score1” and “Score2”.
-class label is called “Result”.
-class label has two possible values “Pass” and “Fail”.
-The data can be divided into two categories or classes:

Classification



If we have some new data, say “Score1 = 25” and “Score2 = 36”, what value should be assigned to “Result” corresponding to the new data; in other words, to which of the two categories or classes the new observation should be assigned?

To answer this question, using the given data alone we need to find the rule, or the formula, or the method that has been used in assigning the values to the class label “Result”.

The problem of finding this rule or formula or the method is the classification problem.

Example 2 of Classification

Can we define breast cancer as malignant or benign based on tumour size

A tumor can be *benign* (not dangerous to health) or *malignant* (has the potential to be dangerous)

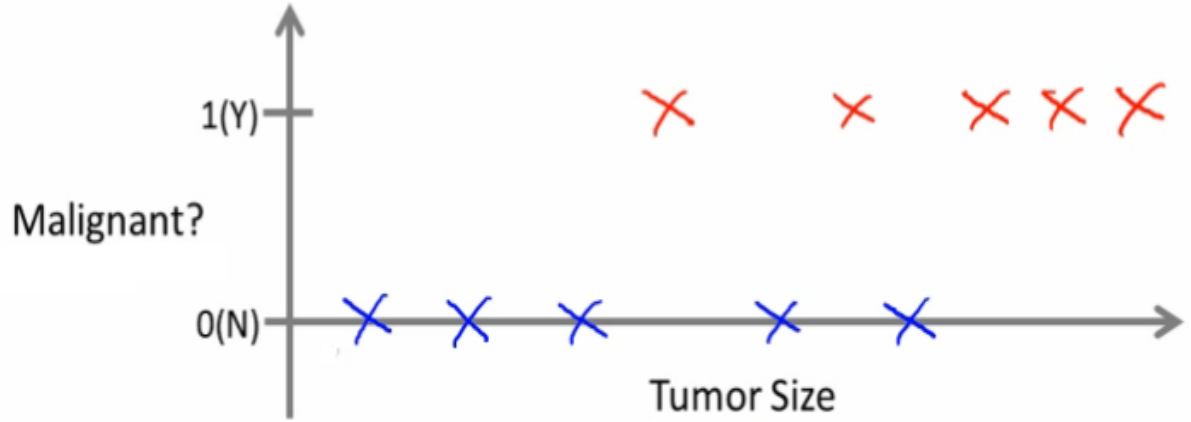
This is an example of a **classification problem**

- Classify data into one of two discrete classes - either malignant or not
- In classification problems, can have a discrete number of possible values for the output

*e.g. maybe have four values

- 0 - benign
- 1 - type 1
- 2 - type 2
- 3 - type 4

Used only one attribute (size)



Example 3 of Classification

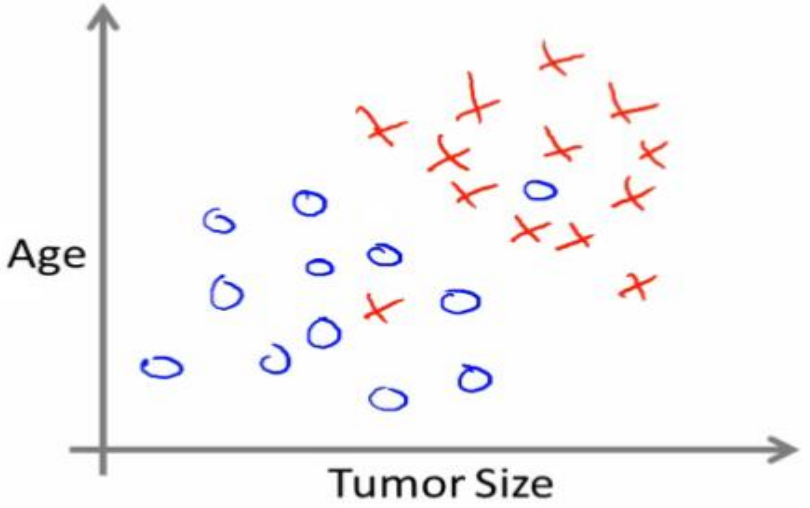
- In other problems may have **multiple attributes** for same problem definition of breast cancer
- We may also, for example, know age and tumor size
- Based on that data, you can try and define separate classes by
    a. Drawing a straight line between the two groups
    b. Using a more complex function to define the two groups (which we'll discuss later)
    c. Then, when you have an individual with a specific

    tumor size and who is a specific age,

    you can hopefully use that information to place

    them into one of your classes

- You might have many features to consider
    a. Clump thickness
    b. Uniformity of cell size
    c. Uniformity of cell shape



Classification: Few Real life examples

Optical character recognition: problem of recognizing character codes from their images, multiple classes.

Face recognition :the input is an image, the classes are people to be recognized, multiple classes.

Speech recognition: the input is acoustic and the classes are words that can be uttered

Medical diagnosis : the inputs are the relevant information about the patient and the classes are the illnesses.

Classification Algorithms

a) Logistic regression

b) Naive Bayes algorithm

c) k-NN algorithm

d) Decision tree algorithm

e) Support vector machine algorithm

f) Random forest algorithm

Regression

In machine learning, a regression problem is the problem of predicting the value of a numeric variable based on observed values of the variable. The value of the output variable may be a number, such as an integer or a floating point value. These are often quantities, such as amounts and sizes. The input variables may be discrete or real-valued.

Regression : Example 1

Consider the data on car prices given in Table

Price (US$)	Age (years)	Distance (KM)	Weight (pounds)
13500	23	46986	1165
13750	23	72937	1165
13950	24	41711	1165
14950	26	48000	1165
13750	30	38500	1170
12950	32	61000	1170
16900	27	94612	1245
18600	30	75889	1245
21500	27	19700	1185
12950	23	71138	1105

Prices of used cars: example data for regression

Suppose we are required to estimate the price of a car aged 27 years with distance 93240 KM and weight 1200 pounds.

This is an example of a regression problem because we have to predict the value of the numeric variable “Price”.

Example 2 of Regression

- How do we predict housing prices
  - Collect data regarding housing prices and how they relate to size in feet

- **Example problem:** "Given this data, a friend has a house 750 square feet - how much can they be expected to get?"

- What approaches can we use to solve this?
  - Straight line through data
    - Maybe $150 000
  - Second order polynomial
    - Maybe $200 000
  - One thing we discuss later - how to chose straight or curved line?
  - Each of these approaches represent a way of doing supervised learning

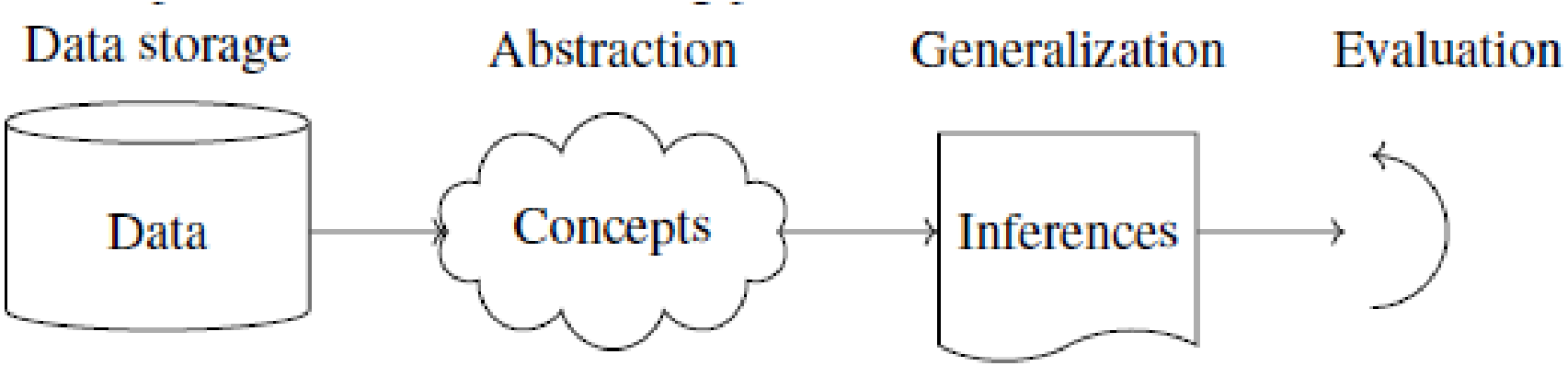
Example 2 of Regression continued….

- *What does this mean?*
  - We gave the algorithm a data set where a "right answer" was provided
  - So we know actual prices for houses
    - The idea is we can learn what makes the price a certain value from the **training data**
    - The algorithm should then produce more right answers based on new training data where we don't know the price already
      1. i.e. predict the price
- We also call this a **regression problem**
  - Predict continuous valued output (price)
  - No discrete values

Basic components of learning process

Definition

A computer program which learns from experience is called a machine learning program or simply a learning program. Such a program is sometimes also referred to as a **learner**.



Basic components of learning process

1. **Data storage:** Facilities for storing and retrieving huge amounts of data are an important component of the learning process. Humans and computers alike utilize data storage as a foundation for advanced reasoning.
2. **Abstraction :** Abstraction is the process of extracting knowledge about stored data. This involves creating general concepts about the data as a whole. The creation of knowledge involves application of known models and creation of new models.

The process of fitting a model to a dataset is known as ***training***. When the model has been trained, the data is transformed into an abstract form that summarizes the original information.

Basic components of learning process

3. **Generalization :** The term generalization describes the process of turning the knowledge about stored data into a form that can be utilized for future action. These actions are to be carried out on tasks that are similar, but not identical, to those what have been seen before. In generalization, the goal is to discover those properties of the data that will be most relevant to future tasks.

4. **Evaluation :** It is the process of giving feedback to the user to measure the utility of the learned knowledge. This feedback is then utilised to effect improvements in the whole learning process.

Understanding data

Unit of observation

By a unit of observation we mean the smallest entity with measured properties of interest for a study.

Examples

- A person, an object or a thing
- A time point
- A geographic region
- A measurement

Examples and features

Datasets that store the units of observation and their properties can be imagined as collections of data consisting of the following:

- **Examples**

An “example” is an instance of the unit of observation for which properties have been recorded.

An “example” is also referred to as an “instance”, or “case” or “record.” (It may be noted that the word “example” has been used here in a technical sense.)

- **Features**

A “feature” is a recorded property or a characteristic of examples. It is also referred to as

“attribute”, or “variable” or “feature.”

Examples for “examples” and “features”

1. Cancer detection

Consider the problem of developing an algorithm for detecting cancer. In this study we note the following.

(a) The units of observation are the patients.

(b) The examples are members of a sample of cancer patients.

(c) The following attributes of the patients may be chosen as the **features**:

- gender
- age
- blood pressure
- the findings of the pathology report after a biopsy

Examples for “examples” and “features”

2.Spam e-mail

Let it be required to build a learning algorithm to identify spam e-mail.

(a) The unit of observation could be an e-mail messages.

(b) The examples would be specific messages.

(c) The features might consist of the words used in the messages.

Examples and features: Representation

Examples and features are generally collected in a “matrix format”. Figure shows such a data set.

Feature Vector
N dimensional
Feature Space

features

year	model	price	mileage	color	transmission
2011	SEL	21992	7413	Yellow	AUTO
2011	SEL	20995	10926	Gray	AUTO
2011	SEL	19995	7351	Silver	AUTO
2011	SEL	17809	11613	Gray	AUTO
2012	SE	17500	8367	White	MANUAL
2010	SEL	17495	25125	Silver	AUTO
2011	SEL	17000	27393	Blue	AUTO
2010	SEL	16995	21026	Silver	AUTO
2011	SES	16995	32655	Silver	AUTO

examples

Different forms of data

1. Numeric data

If a feature represents a characteristic **measured in numbers**, it is called a numeric feature.

2. Categorical or nominal

A categorical feature is an attribute that can take on one of a limited, and usually fixed, number of possible values on the basis of some **qualitative property**. A categorical feature is also called a nominal feature.

3. Ordinal data

This denotes a nominal variable with categories falling in an **ordered list**. Examples include clothing sizes such as small, medium, and large, or a measurement of customer satisfaction on a scale from “not at all happy” to “very happy.”

Examples

In the data given in the previous table., the features “year”, “price” and “mileage” are **numeric** and the features “model”, “color” and “transmission” are **categorical**.

Terminology

Term	Purpose or meaning in the context of Machine learning
Feature, attribute, field, or variable	This is a single column of data being referenced by the learning algorithms. Some features can be input to the learning algorithm, and some can be the outputs.
Instance	This is a single row of data in the dataset.
Feature vector or tuple	This is a list of features.
Dimension	This is a subset of attributes used to describe a property of data. For example, a date dimension consists of three attributes: day, month, and year.
Dataset	A collection of rows or instances is called a dataset. In the context of Machine learning, there are different types of datasets that are meant to be used for different purposes. An algorithm is run on different datasets at different stages to measure the accuracy of the model. There are three types of dataset: training, testing, and evaluation datasets. Any given comprehensive dataset is spilt into three categories of datasets and is usually in the following proportions: 60% training, 30% testing, and 10% evaluation.

Terminology

Term	Purpose or meaning in the context of Machine learning
a. Training Dataset	The training dataset is the dataset that is the base dataset against which the model is built or trained.
b. Testing Dataset	The testing dataset is the dataset that is used to validate the model built. This dataset is also referred to as a validating dataset.
c. Evaluation Dataset	The evaluation dataset is the dataset that is used for final verification of the model (and can be treated more as user acceptance testing).
Data Types	Attributes or features can have different data types. Some of the data types are listed here: • Categorical (for example: young, old). • Ordinal (for example: 0, 1). • Numeric (for example: 1.3, 2.1, 3.2, and so on).
Coverage	The percentage of a dataset for which a prediction is made or the model is covered. This determines the confidence of the prediction model.

Labeled and unlabeled data

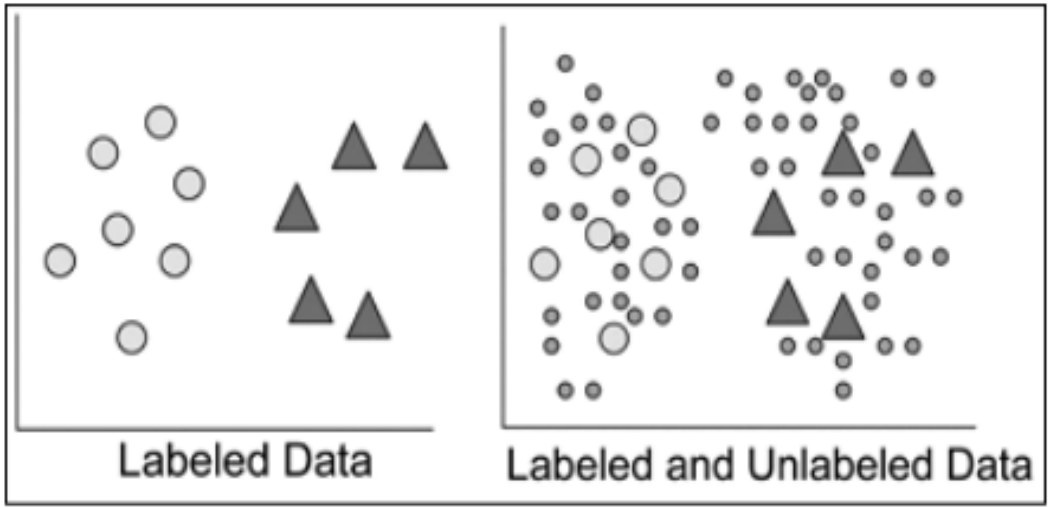
Data in the Machine learning context can either be labeled or unlabeled.Unlabeled data is usually the raw form of the data. It consists of samples of natural or human-created artifacts. This category of data is easily available in abundance. For example, video streams,

audio, photos, and tweets among

others.The unlabeled data

becomes labeled data the moment

a meaning is attached.



Both triangles and bigger circles represent labeled data and small circles represent unlabeled data.

Tasks

A task is a problem that the Machine learning algorithm is built to solve. It is important that we measure the performance on a task.

The term "performance" in this context is nothing but the extent or confidence with which the problem is solved.

Different algorithms when run on different datasets produce a different model. It is important that the models thus generated are not compared, and instead, the consistency of the results with different datasets and different models is measured.

Algorithms

After getting a clear understanding of the Machine learning problem at hand, the focus is on what data and algorithms are relevant or applicable.

There are several algorithms available. These algorithms are either grouped by the learning subfields (such as supervised, unsupervised, reinforcement, semi-supervised, or deep) or the problem categories (such as Classification, Regression, Clustering or Optimization).

These algorithms are applied iteratively on different datasets, and output models that evolve with new data are captured.

Input representation

The general classification problem is concerned with assigning a class label to an unknown instance from instances of known assignments of labels.

In a real world problem, a given situation or an object will have large number of features which may contribute to the assignment of the labels.

But in practice, not all these features may be equally relevant or important.

Only those which are significant need be considered as inputs for assigning the class labels.

These features are referred to as the “input features” for the problem. They are also said to constitute an “**input representation**” for the problem.

More detailed illustration of the supervised learning process

In supervised learning, we are given a labeled training dataset from which a machine learning algorithm can learn a model that can predict labels of unlabeled data points.

we define *h* as the hypothesis,a function that we use to approximate some unknown function

$$f(x) = y; \qquad (1)$$

where x is a vector of **input features** associated with a training example or dataset instance (for example, the pixel values of an image) and y is the outcome we want to predict (e.g., what class of object we see in an image).

More detailed illustration of the supervised learning process

In other words, h(x) is a function that predicts y.

In classification, we define the *hypothesis* function as

$$h : \mathcal{X} \to \mathcal{Y}, \tag{2}$$

where $\mathcal{X} = \mathbb{R}^m$ and $\mathcal{Y} = \{1, ..., k\}$ with class labels k. in the special case of binary classification, we have $\mathcal{Y} = \{0, 1\}$ (or $\mathcal{Y} = \{-1, 1\}$).

And in regression, the task is to learn a function

$$h : \mathbb{R}^m \to \mathbb{R}. \tag{3}$$

Given a training set

$$\mathcal{D} = \{< \mathbf{x}^{[i]}, y^{[i]} >, i = 1, \dots, n\}, \tag{4}$$

we denote the ith training example as $< \mathbf{x}^{[i]}, y^{[i]} >$. Please note that the superscript $[i]$ is unrelated to exponentiation,

More detailed illustration of the supervised learning process

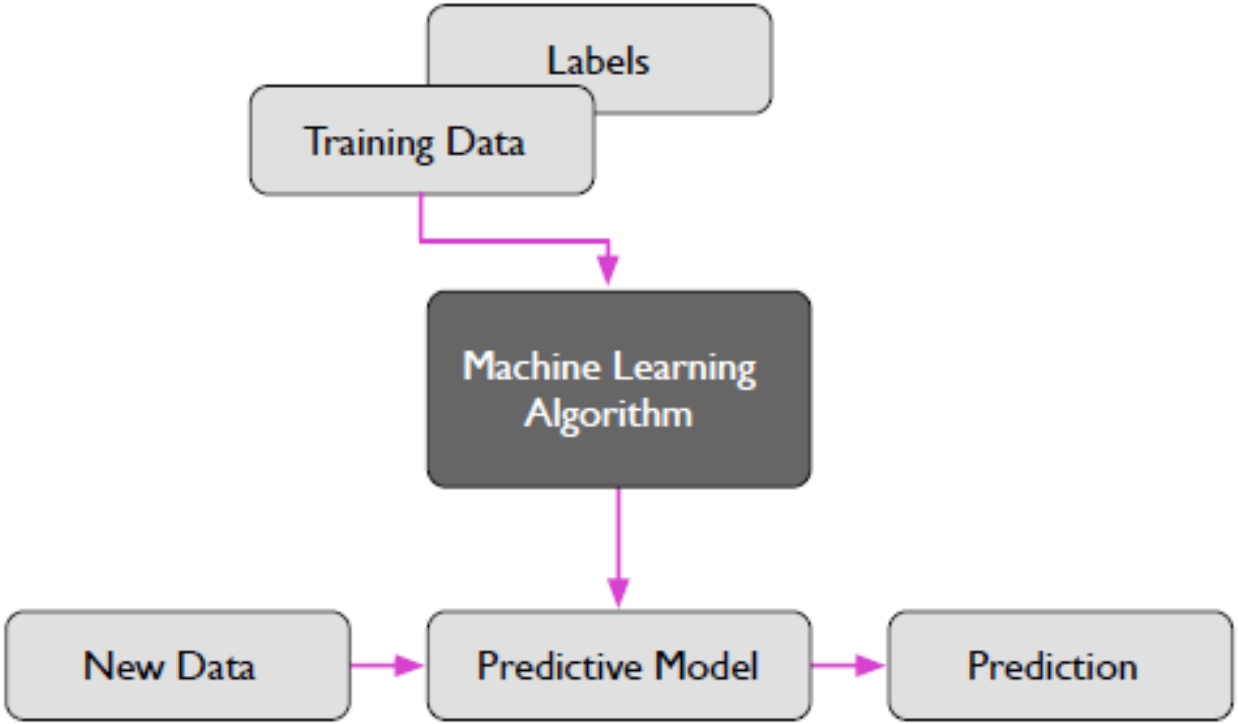


Fig. supervised learning process.

More detailed illustration of the supervised learning process

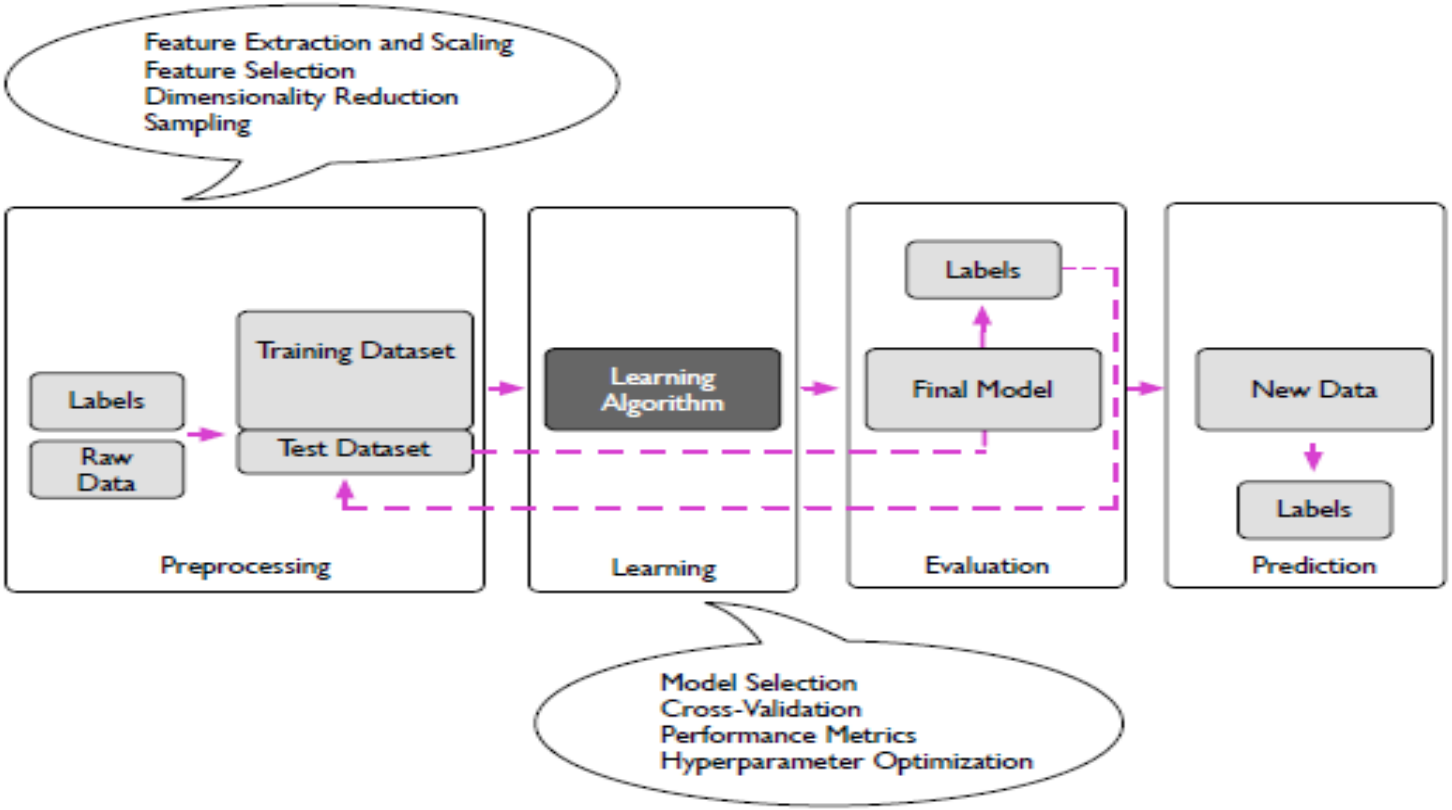


Fig. More detailed illustration of the supervised learning process

Hypothesis

Definition

1. Hypothesis

In a binary classification problem, a hypothesis is a statement or a proposition purporting to explain a given set of facts or observations.

2. Hypothesis space

The hypothesis space for a binary classification problem is a set of hypotheses for the problem that might possibly be returned by it.

Same applicable for multiclass classification

Hypothesis

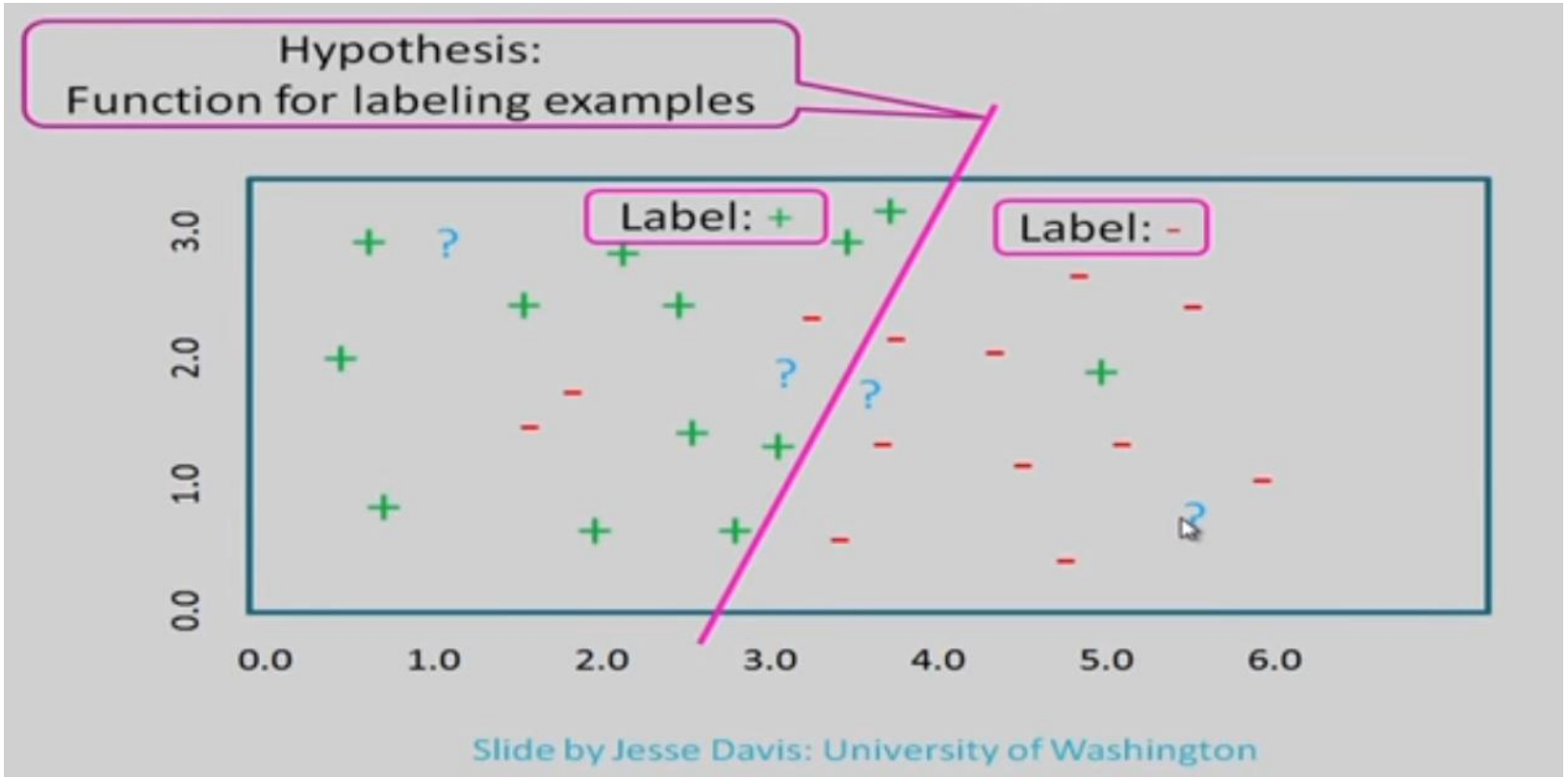
Examples

1. Consider the set of observations of a variable x with the associated class labels given in Table

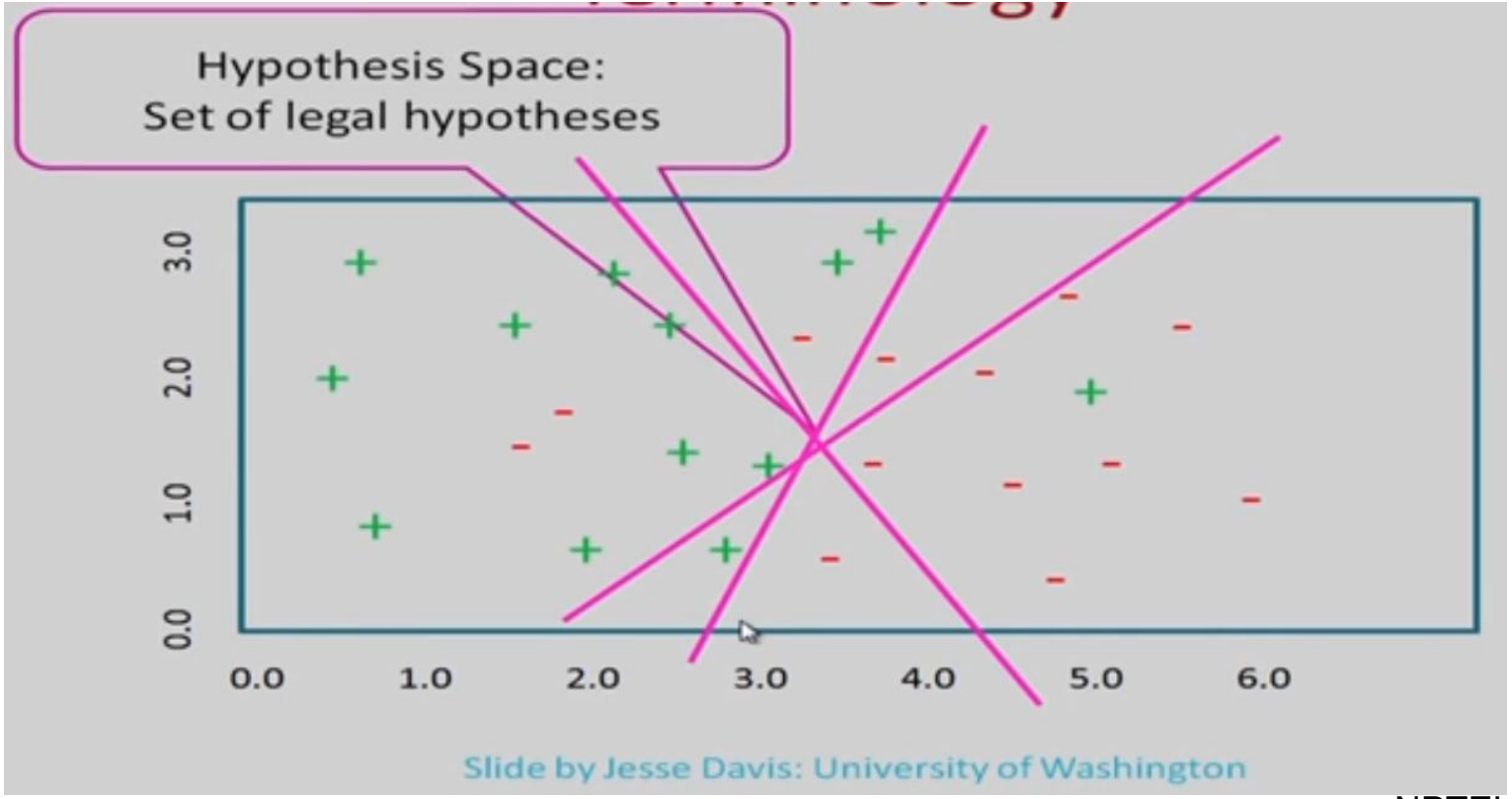
x	27	15	23	20	25	17	12	30	6	10
Class	1	0	1	1	1	0	0	1	0	0

Refer Doc

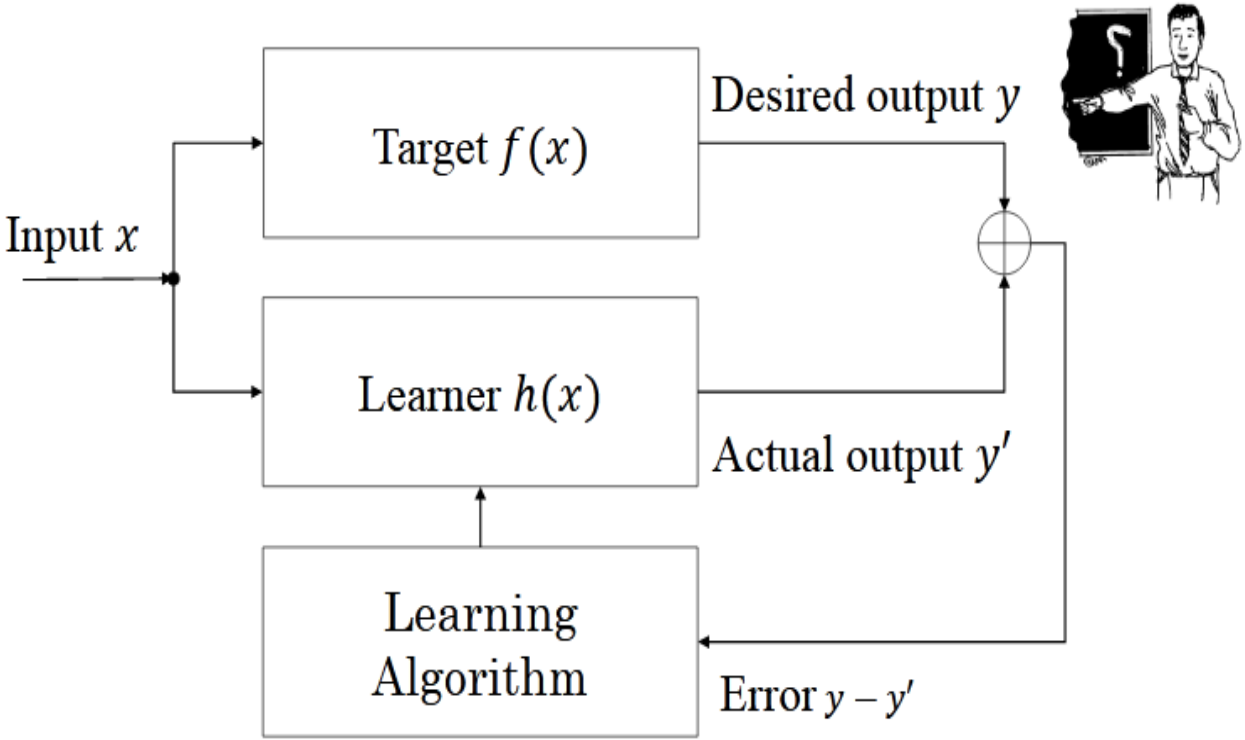
Hypothesis


Hypothesis:
Function for labeling examples
Label: +
Label: -
0.0
1.0
2.0
3.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
Slide by Jesse Davis: University of Washington

Hypothesis Space



Formulation of machine learning



Formulation of machine learning

- Concepts to learn: $X_1, X_2, \ldots, X_{N_c}$

$$X_i = \{x \in X | f(x) = y_i, \quad y_i \in Y\}$$

where $Y = \{y_1, y_2, \ldots, y_{N_c}\}$ is the label set.

- A training data is usually given as a pair (x, y), where x is the observation and y is the label given by a "teacher".
- Learning is the process to find a good "learner" or learning model $h(x)$ to approximate the target function $f(x)$.

A concept is a set of patterns sharing some common properties (e.g. Student, Teacher, etc.)

Formulation of machine learning

- In machine learning, we call $h(x)$ a **hypothesis** . The set of all hypotheses $\mathcal{H}$ is called the **hypothesis space**.
- $\mathcal{H}$ is a set of functions (e.g. all linear functions defined in R^n) when the data are represented as points in R^n.
- Machine learning is an **optimization problem** for finding the best hypothesis $h(x)$ from $\mathcal{H}$, given an observed data set Ω.
- The goodness of a hypothesis can be evaluated by using the following "mean squared error" (MSE) function:

$$E = \frac{1}{|\Omega|} \sum_{x \in \Omega} |f(x) - h(x)|^2$$

- More theoretically, $\mathcal{H}$ is a Hilbert space, and the error can be defined using the norm $|f(x) - h(x)|$.

Formulation of machine learning

- We may use a **loss function** instead of using the error function directly. The simplest loss function is 0-1 loss defined by

$$L = \sum_{x \in \Omega} \mathbf{1}(f(x) \neq h(x))$$

where $\mathbf{1}(P)$ is 1 if P is true, and 0 otherwise.

- The error or loss defined above is **empirical** in the sense that they are defined based on the observed data only. The empirical cost or loss may not be the same as the **predictive value** when we have more data.
- The best predictive error E^* or loss L^* is called the Bayes error or Bayes loss, and the hypothesis $h^*(x)$ that achieves the best error/loss is called the **Bayes Rule**. The goal of machine learning is to find $h^*(x)$ from $\mathcal{H}$.

Issues in Machine Learning

- What algorithms exist for learning general target functions from specific training examples? In what settings will particular algorithms converge to the desired function given sufficient training data? Which algorithms perform best for which types of problems and representations?
- How much training data is sufficient? What general bounds can be found to relate the confidence in learned hypotheses to the amount of training experience and the character of the learner's hypothesis space?
- When and how can prior knowledge held by the learner guide the process of generalizing from examples? Can prior knowledge be helpful even when it is only approximately correct?

Issues in Machine Learning

- What is the best strategy for choosing a useful next training experience, and how does the choice of this strategy alter the complexity of the learning problem?
- What is the best way to reduce the learning task to one or more function approximation problems? Put another way, what specific functions should the system attempt to learn? Can this process itself be automated?
- How can the learner automatically alter its representation to improve its ability to represent and learn the target function?

Steps to solve Machine Learning Problems

Data Gathering

1 *Collect data.* You could collect the samples by scraping a website and extracting data, or you could get information from an RSS feed or an API. You could have a device collect wind speed measurements and send them to you, or blood glucose levels, or anything you can measure. The number of options is endless. To save some time and effort, you could use publicly available data.

Might depend on human work

- Manual labeling for supervised learning.
- Domain knowledge. Maybe even experts.

May come for free, or "sort of"

- E.g., Machine Translation.

The more the better: Some algorithms need large amounts of data to be useful (e.g., neural networks).

The **quantity** and **quality** of data dictate the model **accuracy**

Data Preprocessing

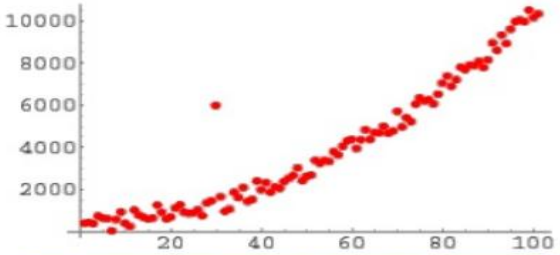
2 *Prepare the input data.* Once you have this data, you need to make sure it's in a useable format. The format we'll be using in this book is the Python list. We'll talk about Python more in a little bit, and lists are reviewed in appendix A. The benefit of having this standard format is that you can mix and match algorithms and data sources.

You may need to do some algorithm-specific formatting here. Some algorithms need features in a special format, some algorithms can deal with target variables and features as strings, and some need them to be integers. We'll get to this later, but the algorithm-specific formatting is usually trivial compared to collecting data.

Is there anything **wrong** with the data?

- Missing values
- Outliers
- Bad encoding (for text)
- Wrongly-labeled examples
- Biased data
  - Do I have many more samples of one class than the rest?

Need to fix/remove data?



Feature Engineering

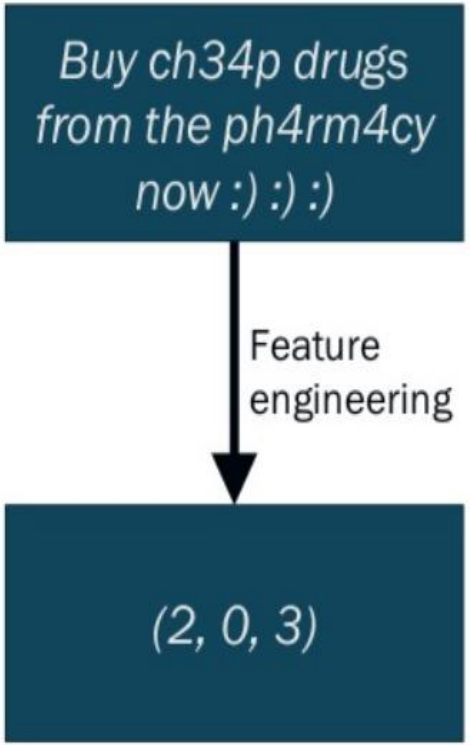
What is a feature?

> *A feature is an individual measurable property of a phenomenon being observed*

Our inputs are represented by a **set of features**.

To classify spam email, features could be:

- Number of words that have been *ch4ng3d* like this.
- Language of the email (0=English, 1=Spanish)
- Number of emojis



Feature Engineering

Extract **more** information from **existing** data, not adding "new" data per-se

- Making it more **useful**
- With good features, most algorithms can learn **faster**

It can be an art

- Requires thought and knowledge of the data

Two steps:

- Variable transformation (e.g., dates into weekdays, normalizing)
- Feature creation (e.g., n-grams for texts, if word is capitalized to detect names, etc.)

3. *Analyze the input data.* This is looking at the data from the previous task. This could be as simple as looking at the data you've parsed in a text editor to make sure steps 1 and 2 are actually working and you don't have a bunch of empty values. You can also look at the data to see if you can recognize any patterns or if there's anything obvious, such as a few data points that are vastly different from the rest of the set. Plotting data in one, two, or three dimensions can also help. But most of the time you'll have more than three features, and you can't easily plot the data across all features at one time. You could, however, use some advanced methods we'll talk about later to distill multiple dimensions down to two or three so you can visualize the data.
4. If you're working with a production system and you know what the data should look like, or you trust its source, you can skip this step. This step takes human involvement, and for an automated system you don't want human involvement. The value of this step is that it makes you understand you don't have garbage coming in.

5 *Train the algorithm.* This is where the machine learning takes place. This step and the next step are where the "core" algorithms lie, depending on the algorithm. You feed the algorithm good clean data from the first two steps and extract knowledge or information. This knowledge you often store in a format that's readily useable by a machine for the next two steps.

In the case of unsupervised learning, there's no training step because you don't have a target value. Everything is used in the next step.

6 *Test the algorithm.* This is where the information learned in the previous step is put to use. When you're evaluating an algorithm, you'll test it to see how well it does. In the case of supervised learning, you have some known values you can use to evaluate the algorithm. In unsupervised learning, you may have to use some other metrics to evaluate the success. In either case, if you're not satisfied, you can go back to step 4, change some things, and try testing again. Often the collection or preparation of the data may have been the problem, and you'll have to go back to step 1.

7 *Use it.* Here you make a real program to do some task, and once again you see if all the previous steps worked as you expected. You might encounter some new data and have to revisit steps 1–5.

Algorithm Selection and Training

Supervised

- Linear classifier
- Naive Bayes
- Support Vector Machines (SVM)
- Decision Tree
- Random Forests
- k-Nearest Neighbors
- **Neural Networks (Deep learning)**

Unsupervised

- PCA
- t-SNE
- k-means
- DBSCAN

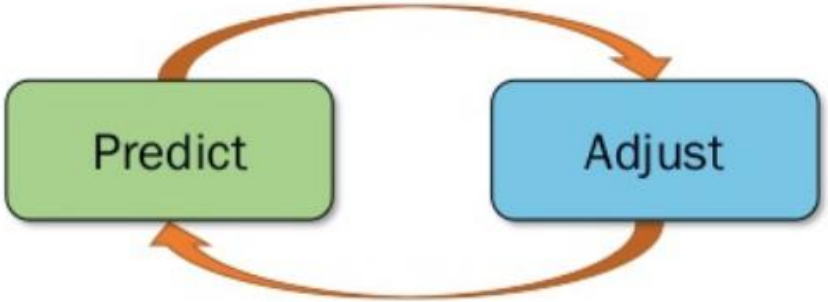
Reinforcement

- SARSA–λ
- Q-Learning

Algorithm Selection and Training

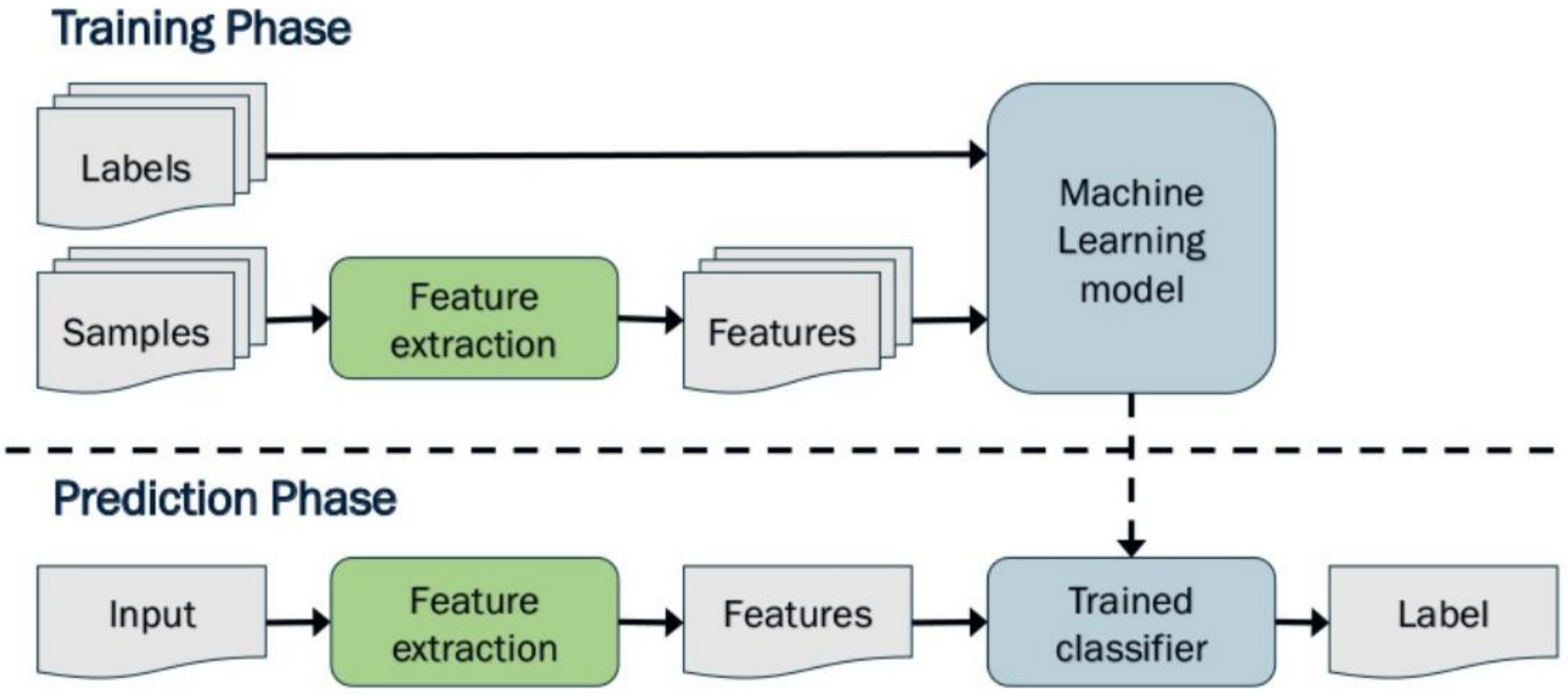
Goal of training: making the correct prediction as often as possible

- Incremental improvement:



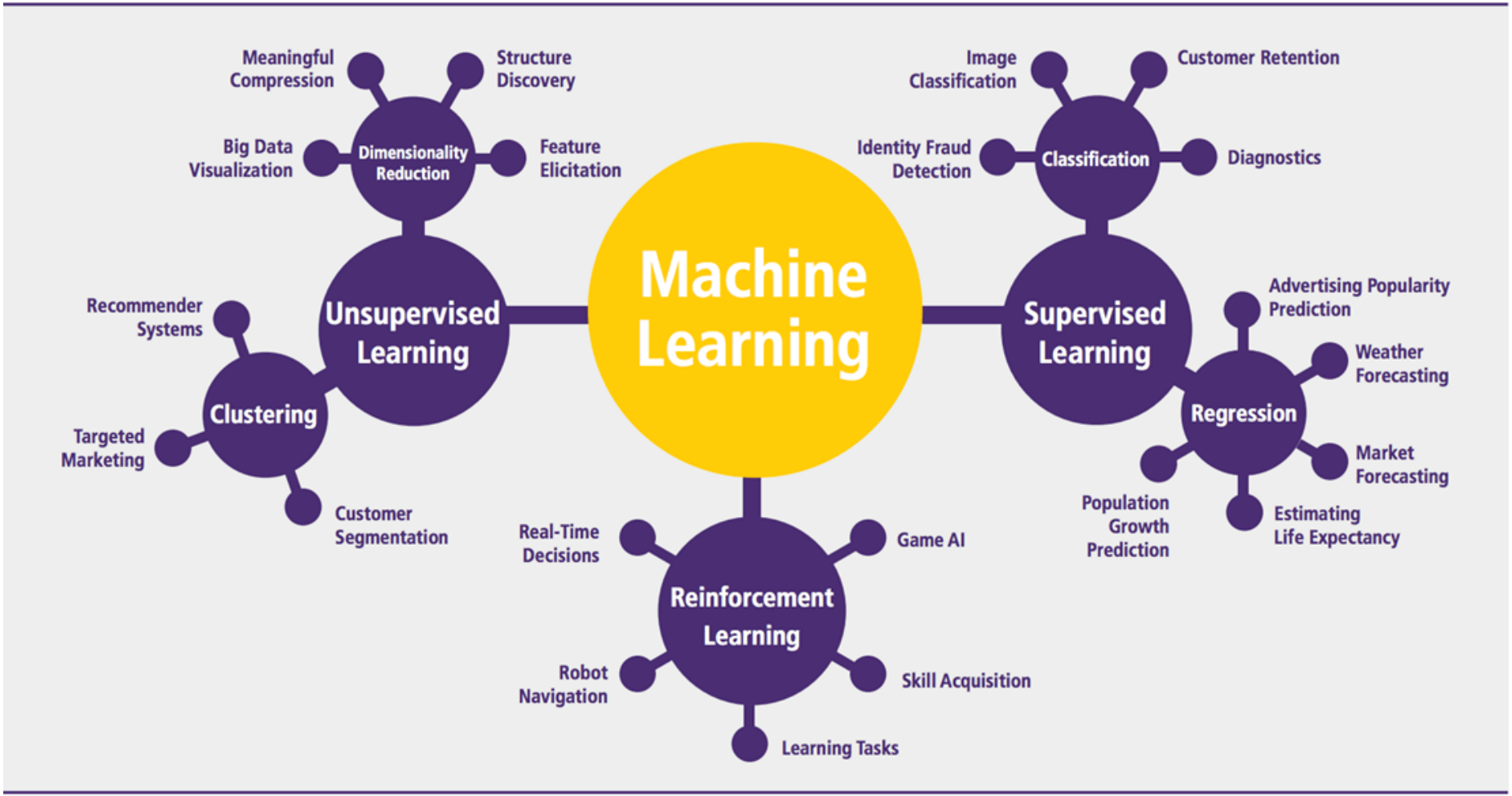
- Use of metrics for **evaluating** performance and comparing solutions
- **Hyperparameter tuning**: more an art than a science

Making Predictions



Applications of machine learning

- Email spam detection
- Face detection and matching (e.g., iPhone X)
- Web search (e.g., DuckDuckGo, Bing, Google)
- Sports predictions
- Post office (e.g., sorting letters by zip codes)
- ATMs (e.g., reading checks)
- Credit card fraud
- Stock predictions
- Smart assistants (Apple Siri, Amazon Alexa, . . .)
- Product recommendations (e.g., Netflix, Amazon)
- Self-driving cars (e.g., Uber, Tesla)
- Language translation (Google translate)
- Sentiment analysis
- Medical diagnoses


Machine Learning
Unsupervised Learning
Dimensionality Reduction
Meaningful Compression
Structure Discovery
Big Data Visualization
Feature Elicitation
Clustering
Recommender Systems
Targeted Marketing
Customer Segmentation
Supervised Learning
Classification
Image Classification
Customer Retention
Identity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Market Forecasting
Population Growth Prediction
Estimating Life Expectancy
Reinforcement Learning
Real-Time Decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks